NETWORK FLOW MODELING FOR TRAFFIC CONTROL

by

Marc Kaltenbach*

* Bishop Univesity, Lennoxville (Québec)

Université de Montréal
Centre de recherche sur les transports - Publication #348
Juillet 1984

Dépôt légal - 3e trimestre 1984

SUMMARY

Here is only one chapter from a larger study on traffic responsive control of traffic signals in urban street networks, by N. Gartner and M. Kaltenbach, to be published by the US Department of Transportation. This chapter is distributed separately in order to provide an easier reference to the traffic simulation model developed by the author. This model is designed to be used for optimizing the control of traffic signal.

Chapter 4 : Network Flow Modeling for Traffic Control Purposes

A model of traffic evolution is proposed as a compromise between two conflicting requirements. On the one hand the model must be realistic enough to permit reasonably accurate measurements of vehicle delays and stops in networks. On the other hand the model must be simple enough for a simulation not to require too much computing effort. Our model has the following characteristics :

1 - It has the form of a dynamical system.

2 - Vehicle flows are represented in aggregated forms.

3 - Storage capacity limits on arcs are enforced.

4 - the simulation work can be performed in a decentralized way and in parallel time by micro-computers periodically exchanging information on a neighbor to neighbor basis.

5 - The model is deterministic. This choice is justified by the following considerations :

a - over short periods of time, the phenomenon of platoon formation at street intersections, as well as the imposition of speed limits, result in some regularity in traffic flows.

b - to a certain extent the stochastic properties of traffic flows can be taken into account when estimating the parameters of the deterministic model.

4.3 Network Topology

Networks considered in this report are of the type represented in Fig. 4.1. Though not all physical street networks are of this type, a large number of practical situations are covered by it; other cases would call for straightforward extensions of current developments. In this model, the arcs between nodes are (street) links indexed 1,2,..., NbLinks joining nodes indexed 1,2,...,NbNodes. The links are classified as either 'horizontal' or 'vertical' for the sole purpose of distinguishing between conflicting link approaches to a node. Indeed, in the physical network to which the proposed model corresponds, the successive links of a street may progressively change orientation. Link type identification is made easy by specifying that the 'horizontal' link indices range from 1 to NbHoriz where NbHoriz is the number of 'horizontal' links. Otherwise the indexing of links and nodes is arbitrary. The set of links belonging to the network is denoted 'LinkSet'. In this exposition I and J represent generic link indices, and N, a generic node index.

Index maps are used to specify a network topology. NextLink[I,d] is the index of the link immediately downstream of link I along direction d. The extremity of link I, by which traffic exits link I, is called the 'end of link I'. At the end of link I:

(i)	if d=-1	a left turn is made
(ii)	if d=0	vehicles proceed straight-on
(iii)	if d=1	a right turn is made.

If NextLink[I,d] = 0 the downstream link along direction d is not part of the network; such links are also called 'Exit links'. In Fig. 4.1, no index is associated with these links. Nextlink[I,d] = -1 if there does not exist a downstream link along direction d. From map NextLink, read as data, the map PrevLink can be inferred; provided NextLink[I,d]>0, PrevLink[I,d] is the link index of the link upstream of link I such that:

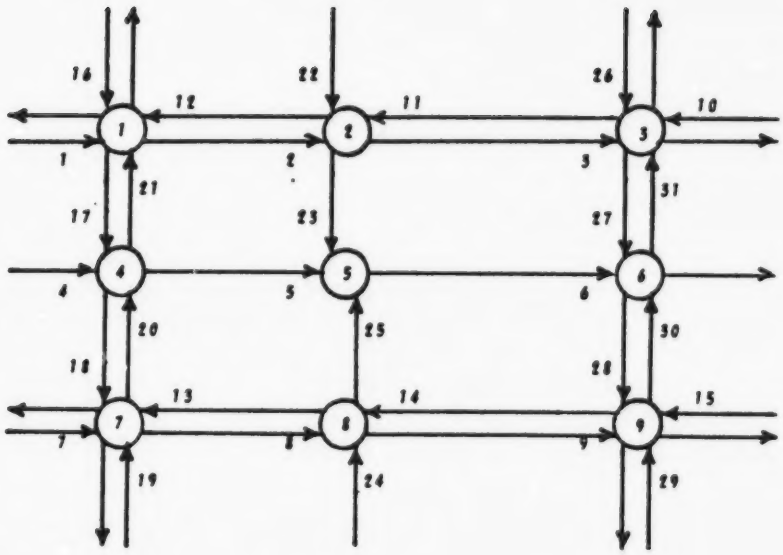

Fig. 4.1 : Network example with 9 nodes and 31 links; turns at nodes are not indicated.

$$\text{PrevLink}[\text{NextLink}[I,d],d] = I \qquad (4.1)$$

If such a link does not exist, set PrevLink[I,d] = 0. Finally, EndNode[I] gives the index of the node at the end of link I. Map EndNode is also read as data.

4.4 Link Description

Links between nodes are assumed to correspond to single lane traffic. Otherwise parallel lanes are aggregated into an equivalent single lane. It is further assumed that link lengths can be partitioned into an integral number of segments of length V*Del called blocks. V corresponds to an average speed (m/sec) of vehicles in free motion over a link. A different V may be specified for every link. Del is a time discretization step size. Choice of a Del Value, for instance Del = 5 seconds, results from a compromise between the desire for accuracy in link length measurement and subsequent optimization difficulties associated with very small Del Values.

The number of blocks in link I is denoted NBlk[I]. The assumed length of link I is therefore NBlk[I]*V*Del meters. A link I that does not have any upstream link belonging to the network is called an 'entrance' link: otherwise it is called an 'inside' link. As it is not necessary to identify blocks on entrance links, an entrance link I is characterized in the model by specifying NBlk[I] = 0. The end point of a link is the point of the link by which vehicles exit the link. A 'network entrance' is the end point of an entrance link.

The blocks over an inside link I are indexed 1,2,...,NBlk[I] in such a way that block NBlk[I] is the block just before the end of link I. That block is referred to later as the 'end block' on link I. These various denotations are summarized in Fig. 4.2.

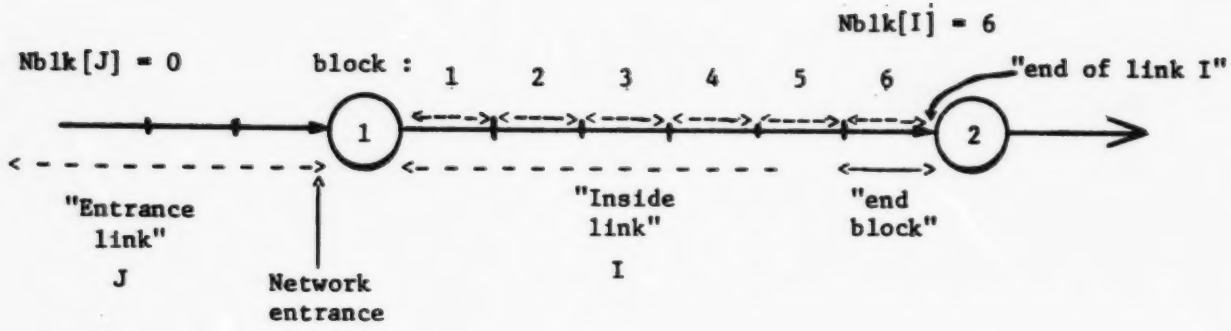

Fig. 4.2 : Link nomenclature.

4.5 Flows on Links

In accordance with the concept of a discrete time system, the distribution of vehicles over the network links is considered only at specific successive time instants, 0, Del,2*Del,...etc. seconds. Thus the reference 'at stage t' denotes the time instant t * Del. The reference 'during stage t' denotes the time interval [(t-1)*Del, t*Del].

At stage t, the distribution of vehicles over link I may be composed of 'free flows' and of a 'queue'. Free flow corresponds to vehicle motion unimpeded by other vehicles or by the absence of right-of-way at an intersection. A queue corresponds to an accumulation of venicles at rest.

The free flow distribution over an inside link I is given by:

$$\text{Flow}[I,k],\ k = 1, \ldots, \text{NBlk}[I] \qquad (4.2)$$

where Flow[I,k] is the average flow rate value, in number of vehicles per seconds, over the points of the kth block of link I; block k = NBlk[I] corresponds to the end block on link I. A distribution of vehicle flow rates over link I at stage t is presented in Fig. 4.3. Consideration of the case of an entrance link is described later.

During a particular stage, vehicles arriving at the end of link I may not have the right-of-way out of link I; then a queue may form in front of node EndNode[I]. Denote by Del*x[I,d] the number of queueing vehicles at EndNode[I], at stage t, due to exit link I along direction d. It will be clear later why we chose to express variable x in flow rate units.

In this model vehicles move in free flow (i.e., at speed V) until the end of a link and only then may join a queue. It can be shown that this way of specifying queue formation is appropriate for computing the sum of vehicle delays in the network. A more

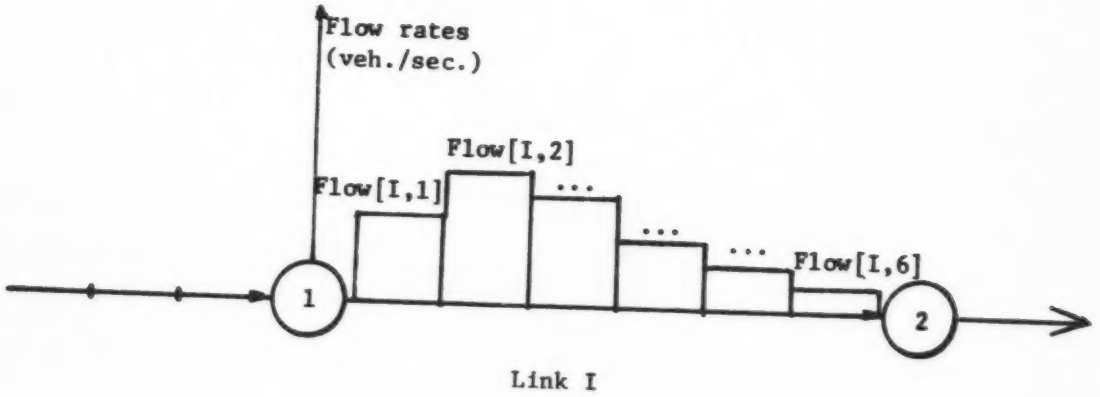

Fig. 4.3 : Flow rate distribution over link I.

sophisticated model of queue formation in which vehicles join a queue by its upstream end is available in [16]. The latter model, which is more appropriate for the measurement of kinetic energy losses, can be superimposed over the present model.

4.6 Link Storage Capacities

An average vehicle in a queue is assumed to occupy VehLgth meters of link length, where VehLgth takes into account the possibility of several lanes in the corresponding physical network link. The link storage capacity is then of NBlk[I]*V*Del/VehLgth vehicles. Since available link storage space at a particular stage is a quantity often referred to in the remaining of this report, it is convenient to introduce an array Sp such that Del*Sp[I] is the available remaining storage capacity, in number of vehicles, on link I at a given stage. Variables Sp are called 'space' variables. We have:

$$Sp[I] = NBlk[I]*V/VehLgth - \sum_{d} x[I,d] \sum_{k=1}^{NBlk[I]} Flow[I,k] \quad (4.3)$$

In this chapter, unless stated otherwise, all summation signs over index d, corresponding to a direction out of a link I, indicate sums over the d values in {-1,0,1} such that NextLink[I,d] > -1. Space is also expressed in flow rate units.

In reality the link space available to vehicles entering link I depends on the position of the upstream end of the queue on link I. A more detailed model of internal queue dynamics [16] would permit a more accurate measurement of the space available on link I at a given stage. A satisfactory compromise between the computational simplicity of the proposed model of queue dynamics and the desire for better link space accuracy might be obtained by expressing Sp[I] as a function of the variables already present in (4.3) and, when there is a queue on link I, of the time duration the end of link I has had the right-of-way, if it has.

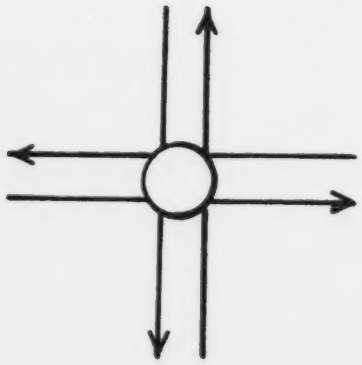

Fig. 4.4 : A simple cross intersection;
right and left turns are allowed.

4.7 Control at Nodes

The case of a simple cross intersection, as in Fig. 4.4 is presented first; possible extensions are indicated next. To the network users the control at street intersections appears as successive green and red durations, i.e., phases. We represent these phases by sequences of '1' or '0' (or equivalently true or false Boolean) values.

Traditionally, the control at a node has been required to conform to Cycle-Split-Offset constraints [14]. In our experiments, this type of control is used for comparison purposes and has the form:

$$u_N[t] = 1 \text{ if } (t + \text{OFFSET}[N]) \text{ MOD CYCLE} < \text{SPLIT}[N]$$
$$= 0 \text{ otherwise} \qquad (4.4)$$

where OFFSET[N], SPLIT[N] are the offset and split control parameters for node N and CYCLE is the cycle value common to all nodes in the network. These control parameters are expressed in step size units. MOD is the modular arithmetic operator; $u_N[t]$ is the control value at node N for the duration of stage t. The notation $u_N[t]$ implies that N = EndNode[I] for some link I in LinkSet.

The right-of-way at a node cannot be given simultaneously to conflicting link directions. In general, a function denoted Rgt(I,$u_N[t]$) is used to determine if vehicles at the end of link I have the right-of-way. Rgt=1 if vehicles have the right-of-way, otherwise Rgt=0. The end of link I has the right-of-way if the following Boolean expression ($I \leq$ NbHoriz) AND $u_N[t]$ is true.

A control policy at node N is a sequence of successive 0-1 control values. Since the traffic light phases are of bounded durations, we require for a control sequence to be admissible that at least L_1 and not more than L_2 consecutive terms in a sequence have the same value, where L_1 and L_2 are appropriate integers and $L_1<L_2$. For example, selecting L_1 = 6, L_2 = 12 and Del = 5, corresponds to specifying phase durations bounded between 30 and 60 seconds. It is also necessary to fully characterize the control applied at a node N at a stage t; i.e., in addition to the control value $u_N[t]$, the duration (in number of stages) of the current phase until stage t (i.e., time t*Del) must be specified. We denote this number $Ph_N[t]$ and call the vector ($u_N[t]$, $Ph_N[t]$) the 'control state' at node N, stage t.

This type of control is more general than the traditional Cycle-Split-Offset control since it does not constrain consecutive phases to a fixed cycle length. Relaxation of this constraint improves the chances of finding a global minimum. We found [16] that with control values at a node allowed to be continuous (i.e., valve type of control) the sum of vehicle delays over a simulation period is a convex function over a convex control space; as a result any local minimum of the performance index is also a global minimum. On the

other hand, numerical experiments with Cycle-Split-Offset 0-1 control values have indicated the presence of many local (i.e., false) minima. In the search for control sequences resulting in a global minimum of vehicle delays we therefore do not wish to introduce unnecessary restraints that may be responsible for some of the local minima.

It may happen that control at a node requires more than two independent phases within a cycle. This would be the case for more complex intersection designs than in Fig. 4.4, and for varying advance green and amber durations. Provided the order of appearance of the successive independent phases is fixed, and these phases rank ordered as $1,2,\ldots,NPh_N$, where NPh_N is the number of independent phases for node N, a natural extension of the current control model consists in redefining variable $u_N[t]$ as the rank order of the incumbent phase during stage t which then can assume NPh_N values. The Rgt function in the traffic simulation model must be modified accordingly and the optimization algorithm must insure that the successive phase durations are bounded between limits now depending on the rank order of the current phase. Such modifications would not add in any significant way to the complexity of the current traffic model. On the other hand, the modeling and control difficulties would increase considerably if some flexibility was allowed in the order of appearance of successive phases. It is questionable, however, to what extent such flexibility would be desirable.

4.8 Service Rates at the End of a Link

In reality street links may have several lanes, the road bed may be in various states of evenness and vehicles parked on street sides as well as the presence of traffic in the opposite direction may create friction on vehicle flow. There are also speed limitations to which vehicles are expected to conform. As a result it is possible to characterize every link I by a maximum flow rate value denoted Fl[I]. Fl[I] must obey consistency bounds:

$$0 \leq Fl[I] < V/VehLgth[V] \qquad (4.5)$$

where VehLgth[V] denotes the effective length occupied by a vehicle in a string of vehicles moving at speed V.

By service rate Serv[I,d] we denote the maximum possible flow rate (Veh/Sec) out of link I along direction d. The maximum service rate actually possible during stage t is: Serv[I,d] * Rgt(I, u_N[t]).

The merging of several flows into a link is a source of modeling difficulties; we must insure that the maximum flow rate into any link J, resulting from the combined service rates of upstream links at any time, does not exceed the upper flow rate limit Fl[J] specified for link J. One solution to this problem is to constrain the service rates. For instance, if S_1 and S_2 are service rates corresponding to simultaneous flows from two links into link J then the estimates for S_1 and S_2, are required to satisfy the constraint:

$$S_1 + S_2 \leq Fl[J] \qquad (4.6)$$

The allocation of values S_1 and S_2 within this limit can be either according to fixed proportions or according to proportions varying as functions of the actual vehicle distributions on the upstream links at the start of the simulation stage during which these service rates apply.

4.9 Discrete Time State Space Model

The system state associated with link I is represented by a vector X_I the components of which are the flow, queue and space variables associated with link I. X_{t-1} denotes a vector obtained by concatenating vectors X_I for all links in LinkSet at stage t-1. Vector U_t represents the control actions for the duration of stage t at all nodes in the network. Vector Q_t stands for the externally specified arrival input flow rates at the network entrances for the duration of stage t. We seek a state space representation of traffic evolution of the form:

$$X_t = F[X_{t-1}, U_t, Q_t] \quad t = 1,2,... \qquad (4.7)$$

System function F is realized through two consecutive 'passes' over all network links. The first pass consists of queue updates; the second pass updates space variables and flow rate distributions over links. Decomposing the system state transition operations into two 'passes' offers a convenient way of solving the problem of precedence relationships between link flow distribution updates.

It is computationally advantageous to eliminate explicit references to the time step Del when evaluating the state transition function F; this is why all state variables in the model are expressed in flow rate equivalent units (Veh/Sec).

PASS 1: QUEUE UPDATE

Flow conservation at node I, along direction d, over stage t requires that:

$$x[I,d] = x[I,d] + Q^{in}[I,d] - Q^{out}[I,d] \qquad (4.8)$$

where Q^{in} and Q^{out} correspond to flow in and flow out of node EndNode[I], along direction d, during stage t. The equal sign stands for 'replaces'.

To obtain the flow in, Q^{in} we distinguish between whether or not link I is a network entrance link. If link I is an entrance link (this is ascertained by checking condition NBlk[I] = 0) then the flow rate on the end block of link I at time (t-1)*Del is specified externally. In our computer experiments we use periodic arrivals specified as follows:

$$Q^{in}[I,d] = \text{FracFlow}[I,d] * \text{STO}[I, (t \text{ MOD } (p-1)) + 1] \qquad (4.9)$$

where MOD is the Modulo Operator and p is the period of arrivals in number of time steps. STO[I,n], n=1,...,p is an array storing flow

rate values for one period. FracFlow[I,d] is the proportion of vehicle arrivals allocated to direction d at the node. The FracFlow values are specified as constant model parameters.

If link I is not an entrance link

$$Q^{in}[I,d] = \text{FracFlow}[I,d] * \text{Flow}[I,\text{NBlk}[I]] \qquad (4.10)$$

To obtain the flow out, Q^{out} three cases are distinguished:

(i) Q^{out} cannot exceed the specified service flow rate at the end of link I, i.e.,

$$Q^{out}[I,d] \leq \text{Serv}[I,d] * \text{Rgt}(I, u_N[t]) \qquad (4.11)$$

(ii) Q^{out} cannot exceed the demand for passage at the node:

$$Q^{out}[I,d] \leq x[I,d] + Q^{in}[I,d] \qquad (4.12)$$

(iii) Q^{out} cannot exceed the downstream link storage capacity. Let J = NextLink[I,d], J > 0. We must determine the (Vehicle storage) space available on downstream link J. If there were no vehicle entering link J during stage t, the space available on link J, at the end of stage t, would be:

$$\text{Sp}[J] + \sum_d Q^{out}[J,d]$$

That space also represents the maximum number of vehicles (in flow rate equivalent units) that could enter link J during stage t. The space available, assumed when computing the maximum possible flow into link J, is artificially restricted to Sp[J]. Thus link J storage capacity is underestimated by

$$\text{Del} * \sum_d Q^{out}[J,d] \quad \text{vehicles,}$$

a small quantity controlled by the choice of Del. This underestimation has the advantage of decoupling link update operations which can now be performed in parallel by a corresponding number of independent processors, one for each link. Also, it partially compensates for the overestimations resulting from not calculating the exact position of the end point of the queue.

Over the duration of stage t there may be r different input flow streams into link J (in the case of Fig. 4.4, r = 1 or 2). We must insure that their combination does not violate link J storage capacity. Accordingly we specify:

$$Q^{out}[I,d] \leq FracSp[I,d] * Sp[J] \qquad (4.13)$$

where FracSp[I,d] is the average proportion of the space on link J available to traffic from link I during stage t when link J experiences saturation (i.e., when the sum of all incoming flows exceeds link J storage capacity). The space available then corresponds to the space vacated by vehicles leaving the link during the previous stage. The FracSp values corresponding to simultaneous flows into link J must sum up to 1. These values may be specified either as constants or as functions of actual traffic distributions over upstream links at stage t-1.

The various cases for $Q^{out}[I,d]$ may be summarized as follows:

$$Q^{out}[I,d] = Min\{x[I,d] + Q^{in}[I,d], \\ Serv[I,d]*Rgt(I,u_N[t]), FracSp[I,d]*Sp[J]\} \qquad (4.14)$$

Now x[I,d] is updated according to (4.8). Intermediate variables $Q^{out}[I,d]$ are saved in memory until the end of Pass 2.
Pass 1 terminates when the above queue updates have been performed for all links in LinkSet.

PASS 2: FLOW AND SPACE VARIABLE UPDATES

Vehicle movement during one stage, over link I, is achieved by translating flow rate variables from one block to the next in the following order:

$$\text{Flow}[I,k] = \text{Flow}[I,k-1] \quad \text{for } k = \text{NBlk}[I] \text{ Down to } 2. \qquad (4.15)$$

The new value for Flow[I,1] is obtained by summing all flows entering link I during stage t.

$$\text{Flow}[I,1] = \sum_d Q^{out}[J,d] \qquad (4.16)$$

In this particular case J = PrevLink[I,d] and the summation runs over all d, d ε {-1,0,1}, such that J > 0.

Most of the computing work associated with these update operations can be avoided by storing the flow rates in a ring data structure. On the other hand, a smoothing formula, or a redistribution of flow rates among adjacent blocks, could serve to model the phenomenon of platoon dispersion. For an inside link I, the equation for link space update is:

$$\text{Sp}[I] = \text{Sp}[I] - \text{Flow}[I,1] + \sum_d Q^{out}[I,d] \qquad (4.17)$$

Pass 2 terminates when the above operations have been completed for all links in LinkSet.

4.10 Equivalent Model with Time Delays

The model formulated so far is appropriate for simulating flows in large networks. It is an important component of our proposed on-line traffic control system. We also use it in simulation studies to provide the environment in which to test alternative control system designs. This form of the model is however less suitable for the purpose of node control optimization. The control approach described

below calls for the simultaneous availability of successive system states over a simulation horizon of T time steps. Storing and retrieving these states is very costly in terms of memory requirements and computing time. An equivalent system formulation involving time delays solves this problem. We call this form of the model the 'delayed system', and the other form, the 'non-delayed system'. We shall use the expression 'at step t0' to denote the time (t + t0)*Del; the expression 'during step t0' to denote the time interval ((t + t0 - 1)*Del, (t + t0)*Del]. For the time being, the delayed system is used to compute and store the system state at steps: t0 = 1, ..., T.

The delayed system is obtained by appending a time index to the non-delayed system 'queue' and 'space' variables and recognizing that the flow arriving at the end of an inside link is the flow that entered the link some time earlier. Otherwise the two models are identical. The flow rate values into link I during the successive time steps t0 = 1, ..., T are denoted Flot[I,t0]. A free flow distribution over link I of the delayed system at step t0 is given by Flot[I,k], k = t0,t0-1, ..., t0-NBlk[I]+1. An example of that distribution, presented in Fig. 4.5, is the equivalent of the non-delayed model distribution in Fig. 4.3.

The delayed system equations are listed next. We assume that the initial conditions for the delayed system are equal to the system and control states of the non-delayed system at stage t. The delayed system control sequences are indexed from 1 to T. The delayed system is simulated by repeating the following 'Pass 1' and 'Pass 2' for steps t0 = 1,..., T.

PASS 1: For all links in LinkSet, update the queues according to the flow conservation equation:

$$x[I,d,t0] = x[I,d,t0-1] + Q^{in}[I,d,t0] - Q^{out}[I,d,t0] \quad (4.18)$$

If I is an entrance link:

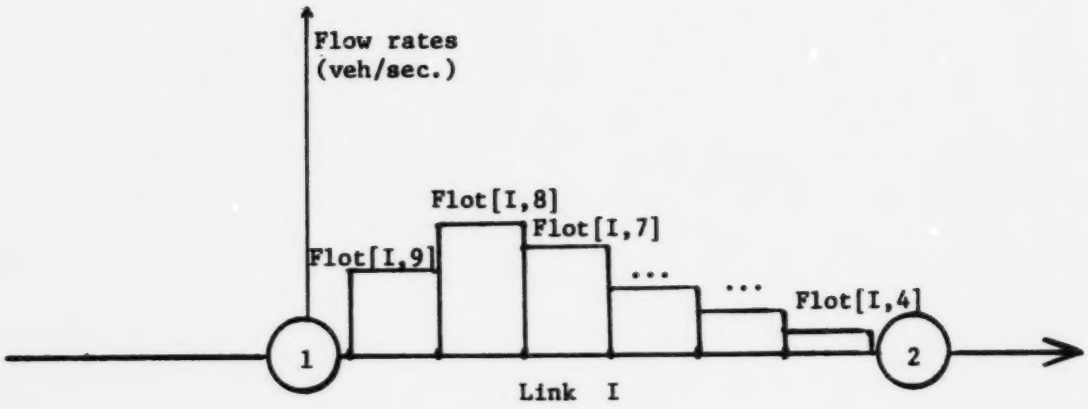

Fig. 4.5 : Flow distribution over link I for the delayed system at time step 9.

$$Q^{in}[I,d,t0] = FracFlow[I,d]*STO[I, ((t+t0) \text{ MOD } (p-1)) + 1] \quad (4.19)$$

If I is an inside link and, $t0 < t + 1$,

$$Q^{in}[I,d,t0] = FracFlow[I,d]*Flow[I,NBlk[I] + 1 - t0] \quad (4.20)$$

If I is an inside link and, $t0 \geq t + 1$,

$$Q^{in}[I,d,t0] = FracFlow[I,d]*Flot[I, t0 - NBlk[I]] \quad (4.21)$$

$$Q^{out}[I,d,t0] = \text{Min } \{x[I,d,t0-1] + Q^{in}[I,d,t0], Serv[I,d]*Rgt(I,u_N[t0]),$$

$$FracSp[I,d]*Sp[NextLink[I,d],t0-1]\} \quad (4.22)$$

PASS 2: For every inside link I in LinkSet, we update the flow into link I

$$Flot[I,t0] = \sum_d Q^{out}[J,d,t0] \quad (4.23)$$

where J = PrevLink[I,d] and the summation runs over all d, $d \in \{-1,0,1\}$, such that $J > 0$.

We update the space variable

$$Sp[I,t0] = Sp[I,t0-1] - Flot[I,t0] + \sum_d Q^{out}[I,d,t0] \quad (4.24)$$

4.11 Choice of a Performance Index

Various practically significant performance indices may be associated with a T time steps network flow simulation. The control approach to be described later requires only that the performance index be step by step additive, i.e., that if H(t0) is a non negative performance measure associated with a system state transition at step t0, the performance associated with the simulation be of the form:

$$PI[T] = \sum_{t0=1}^{T} H(t0) \qquad (4.25)$$

This requirement is met by two common traffic control objectives, namely the minimization of the sum of all vehicle delays and the minimization of the sum of all kinetic energy losses in the network over the simulation period.

We shall restrict our attention to the case of vehicle delays. Except for a few modeling extensions, the case for kinetic energy loss is identical. For a simulation horizon of T time steps, the Total Delay in the network (in vehicle intervals) is equal to:

$$TD[T] = \sum_{t0=1}^{T} \sum_{I} \sum_{d} x(I,d,t0) \qquad (4.26)$$

When T is very large the Average Total Delay per stage may be a more significant measure ATD[T] = TD[T]/T.

The following statement of an open loop optimal control problem concludes this general presentation of network traffic flow models for control purposes. Frequent references to that problem will be made in the next section.

(OPEN-LOOP) OPTIMAL CONTROL PROBLEM:

Given initial system and control states at step 0, and given distributions of traffic arrivals at the network entrances for T time steps in the future, find a set of admissible control sequences, one for each node, that minimizes PI[T].

REFERENCE

[16] M. Kaltenbach, Optimal Control of Time-Varying Flows in networks. RAIRO - recherche opérationnelle - Vol. 15, No. 4, Nov. 1981.